MATEC Web of Conferences **42**, 05001 (2016)
DOI: 10.1051/matecconf/20164205001

A Comparison of Heuristics with Modularity Maximization Objective using Biological Data Sets

Harun Pirim [1]

[1] *Systems Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran, KSA*

Abstract. Finding groups of objects exhibiting similar patterns is an important data analytics task. Many disciplines have their own terminologies such as cluster, group, clique, community etc. defining the similar objects in a set. Adopting the term community, many exact and heuristic algorithms are developed to find the communities of interest in available data sets. Here, three heuristic algorithms to find communities are compared using five gene expression data sets. The heuristics have a common objective function of maximizing the modularity that is a quality measure of a partition and a reflection of objects' relevance in communities. Partitions generated by the heuristics are compared with the real ones using the adjusted rand index, one of the most commonly used external validation measures. The paper discusses the results of the partitions on the mentioned biological data sets.

1 Introduction

Clustering problem has attracted attentions of researchers for decades. The problem exists in almost every discipline and the solutions to address the problem are applicable to any discipline such as engineering, natural and social sciences. While there are general clustering algorithms, it is also desirable to design clustering approaches for specific problems.

In this regard, optimization discipline has unique tools for clustering. Clustering problem can be perceived as an optimization problem where there is a specific objective function to optimize and some constraints to satisfy.

Mathematical programming approaches for clustering exist. However, the mathematical models are practical and applicable to small scale data. Hence, heuristic algorithms are required to solve large scale optimization problems.

Clustering is an unsupervised learning task/tool encountered in various data mining applications. The applications span many fields including physics, astronomy, and bioinformatics employing a plethora of algorithms.

Different disciplines favor distinct terms for a set of similar objects, namely cluster, clique, group, or community. Although there is not a universal definition for the best community, researchers agree that the objects in a community must exhibit similar patterns or must be strongly connected based on a defined relationship. In other words, the similarity of objects within a cluster should be maximized, and the similarity of objects between clusters should be minimized.

Bioinformatics is still a developing field to address life sciences problems using computational techniques. Clustering approaches are employed for data analysis in bioinformatics as well. Analysis of microarray gene co-expression data is one of the bioinformatics problems where clustering is utilized.

Clustering gene expression data is an integrated task that comprises low-level and high-level analysis. Three main steps of cluster analysis for gene expression data are as follows:

1. data pre-processing: preparing the data so that the clustering algorithm can make use of it as an input;
2. employing a clustering algorithm with an appropriate distance measure (if necessary); and
3. using a validation measure (internal and/or external) to validate the quality of the clusters found.

Keeping in mind the importance of analysis of high dimensional gene expression data sets, heuristics are promising approaches for clustering high-throughput data such as the ones generated by microarrays. Microarrays measure expression levels of ten thousands of genes simultaneously in a single chip. Measurements involve relative expression values of each gene through an image processing task.

There is no best clustering approach for the problem on hand and the clustering algorithms are biased towards certain criteria. In other words, a particular clustering approach has its own objective and assumptions about the data.

For example, K-means algorithm is sensitive to noise that is inherent in gene expression data. In addition, the solution (i.e. the final clustering) that the K-means algorithm finds may not be a global optimum since it relies on randomly chosen initial objects.

Hierarchical clustering algorithms are "greedy" which often means that the final solution is suboptimal due to locally optimal choices being made in initial steps, which turn out to be poor choices with respect to the global solution.

Here, three heuristic community structure finding algorithms are employed on five different gene expression data sets. The algorithms have the common objective of maximizing a community defining measure called modularity. The higher modularity values indicate better clustering.

Maximum modularity values generated by the algorithms on the data sets are reported. The partitions by the algorithms are evaluated comparing with the real partitions through a widely used external validation index, adjusted rand index.

The paper is organized as follows: section two describes the community structure finding problem using modularity, section three presents the algorithms and the data sets as well as the results of the study, and section four is the conclusion.

2 Community Structure Finding

There are many community structure finding (also could be mentioned as pattern recognition or clustering) algorithms using modularity maximization on a given network G = (V,E) with m =| E | edges and n =| V | nodes. From an optimization perspective, the problem of finding the best (optimal) community can be modeled as an integer linear programming (ILP) problem. The corresponding ILP is as follows:

$$\max \frac{1}{2m} \sum_{(u,v)\in v^2} (E_{uv} - \frac{\deg(u)\ \deg(v)}{2m}) x_{uv}$$

Subject to

$$x_{uu} = 1, \forall u \quad (1)$$
$$x_{uv} = x_{vu}, \forall u, v \quad (2)$$
$$x_{uv} + x_{vw} - 2x_{uw} \leq 1, \forall u, v, w \quad (3)$$
$$x_{uw} + x_{uv} - 2x_{vw} \leq 1, \forall u, v, w \quad (4)$$
$$x_{vw} + x_{uw} - 2x_{uv} \leq 1, \forall u, v, w \quad (5)$$
$$\forall x_{uv} \in (0,1) \quad (6)$$

x_{uv} are binary variables being 1 if there is a connection between nodes u, v and 0 otherwise. First set of constraints are reflectivity constraints. Second set of constraints are symmetry constraints meaning that if object u is connected to object v then the object v is connected to the object u. Third, fourth, and fifth set of constraints are transitivity constraints meaning that when the object u is connected to the object v and the object v is connected to the object w then the object u is also connected to the object w. Noticing the redundancies in terms of variables and constraints, the number of variables and constraints are of $\binom{n}{2}, \binom{n}{3}$ respectively as the redundancies are removed.

In their paper, Brandes et al.[1] prove the problem to be NP-complete. Hence, heuristic algorithms are required to generate reasonable results close to global optimum. Newman and Girvan [2] propose one of the most cited community structure finding algorithms. The algorithm finds communities removing the most between edges of the graph that is constructed from a data set. One way to define the betweenness is through counting the number of shortest paths passing along an edge. The algorithm's worst-case time complexity is $O(m^2n)$.

Pons and Latapy [3] compute communities using random walks. The algorithm, called walktrap, has worst-case time complexity of $O(mn^2)$. Clauset and Newman [4] propose a fast greedy community structure finding algorithm based on hierarchical agglomeration. The algorithm's worst-case time complexity is O(mdlogn) where d is the depth of the dendrogram describing the community structure. When the network is sparse and dendrogram is balanced, the algorithm runs in $O(n\log^2 n)$ time.

Clique percolation [5] and label propagation [6] are some of the recent methods to find community structures. Furtunato [7] presents a recent review on algorithmic methods to detect community structure in networks.

3 Methods and Results

Three community structure finding algorithms, namely betweenness [2], walktrap [3], greedy [4] are compared using five gene expression data sets. The reason of selecting these algorithms is the ease of implementation and prevalence of these algorithms. R programming implementations of igraph library [8] are employed.

The data sets are summarized in Table 1. The BreastA is a two-channel oligonucleotide microarray data set. The BreastB is one-channel microarray data set. Both are cancer diagnosis data sets. DLBCLA is a diffuse large B-cell lymphoma data set. These three data sets are published in [9]. Leukemia data set is obtained online at http://www.broadinstitute.org/cgibin/cancer/datasets.cgi.

Table 1. Data Sets.

Data Sets	# of objects	#of features	# of classes
BreastA	98	1213	3
BreastB	49	1213	4
DLBCLA	141	661	3
Leukemia	248	985	6
CNS	112	9	4

CNS data set has the nine time points observation of 112 rat genes. The data set is addressed in [10].

Each data set is represented as a complete network where nodes represent the samples (tissues) or the genes (for CNS data only), the edges represent the relationships with Pearson correlation valued strengths. Then the networks are trimmed removing the edges with the least correlation values until the networks become disconnected. The corresponding threshold values for the data sets in the order shown on Table 1 are 0.293, 0.545, 0.839, 0.225, 0.610. The community structure finding algorithms use the trimmed networks to generate partitions and modularity values. The partitions are used to calculate the adjusted rand index [11] values. The application work flow is shown in Figure 1.

The results shown on Table 2 are the maximum modularity values corresponding to the partitions generated by the algorithms. mod1, mod2, mod3 are maximum modularity values obtained by betweenness, walktrap, and greedy algorithms respectively. The values in bold are the maximum of the maximum modularity values found by the algorithms.

The results shown on Table 3 are adjusted rand index values corresponding to the partitions generated by the algorithms. rand1, rand2, rand3 are adjusted rand index values obtained by betweenness, walktrap, and greedy algorithms respectively. The values in bold are the maximum of the maximum modularity values found by the algorithms.

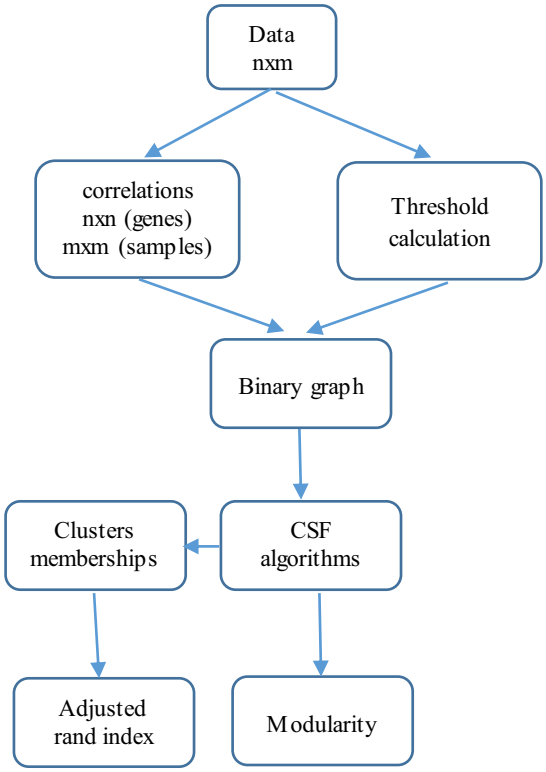

Figure 1. The work flow.

Table 2. Modularity Results.

Data Sets	mod1	mod2	mod3
BreastA	0.436	0.437	0.439
BreastB	0.001	0.004	0.020
DLBCLA	0.001	0.040	0.055
Leukemia	0.423	0.490	0.452
CNS	0.076	0.235	0.238

Adjusted rand index (ARI) values are computed in R using clues [12] package. The adjusted rand index values are calculated by the partition from a clustering algorithm (P1) and the real partition (P2). The ARI(P1,P2) formulation is as follows:

$$\frac{\sum_{i,j}\binom{n_{i,j}}{2} - \left[\sum_i \binom{n_{i.}}{2}\sum_j \binom{n_{.j}}{2}\right]/\binom{n}{2}}{0.5\left[\sum_i \binom{n_{i.}}{2} + \sum_j \binom{n_{.j}}{2}\right] - \left[\sum_i \binom{n_{i.}}{2}\sum_j \binom{n_{.j}}{2}\right]/\binom{n}{2}} \quad (7)$$

$n_{i,j}$ is the number of common objects from clusters i and j, i is the cluster index of the first partition and j is the cluster index of the second partition. $n_{i.}$ represents the number of objects in cluster i. Higher adjusted rand index values indicates better clusters. In other words, the higher the index value the closer the partition to the real partition. ARI values lie between -1 and 1.

Table 3. Rand Index Results.

Data Sets	rand1	rand2	rand3
BreastA	0.856	0.870	0.866
BreastB	0.361	0.673	0.576
DLBCLA	0.391	0.641	0.624
Leukemia	0.817	0.972	0.834
CNS	0.640	0.621	0.534

Figure 2 illustrates the partition obtained by the walktrap algorithm for Leukemia data set. Shaded regions (different colored) indicate the clusters found by the algorithm.

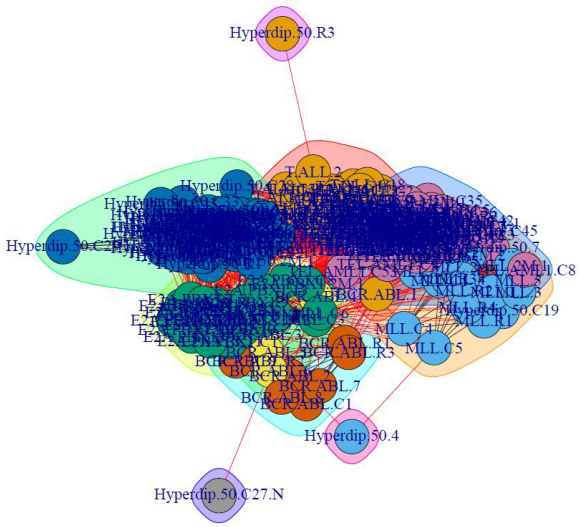

Figure 2. The partition by the walktrap algorithm for Leukemia data set.

4 Conclusion

Performances of three community structure finding algorithms are tested using five gene expression data sets. Maximum modularity and adjusted rand index values are used as performance metrics.

The greedy algorithm finds the highest modularity values except for Leukemia data set. The betweenness algorithm finds the worst modularity values. The walktrap algorithm finds the highest adjusted rand index values except for CNS data set. The betweenness algorithm finds the best rand index value for CNS data. This observation brings the question if the betweenness algorithm works better for the gene clustering since CNS is the only gene clustering data set. The other data sets are all sample (tissue) clustering data sets. Another observation is that there is no clear relationship between maximum modularity values and adjusted rand index values.

References

1. U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, and D. Wagner, IEEE Transactions on Knowledge and Data Engineering, **20**(2):172 – 188, 2010
2. M. E. J. Newman and M. Girvan, Physical Review E, **69**(026113), 2004
3. T. Luigi, D. Luca, P. Pons and M. Latapy, *Lecture Notes in Computer Science*. Springer Berlin, Heidelberg, 2005
4. A. Clauset, M. E. J. Newman, and C. Moore, Physical Review E, **70**(066111), 2004
5. I. Derenyi, G. Palla, and T. Vicsek, Phys. Rev. Lett., **94**(160202), 2005
6. G. Tibely and J. Kertesz, Physica A: Statistical Mechanics and its Applications, **387**(4982-4):19 – 20, 2008
7. S. Fortunato, Physics Reports, **486**:75 – 174, 2010
8. G. Csardi and T. Nepusz, InterJournal Complex Systems, **1695**, 2006
9. Y. Hoshida, J. Brunet, P. Tamayo, T. Golub, and J. Mesirov, PLoS ONE, **11**(e1195), 2007
10. S. Bandyopadhyay, A. Mukhopadhyay, and U. Maulik, Bioinformatics, **23**(21):2859 – 2865, 2007
11. L. Hubert, P. Arabie, J. Class. **2:**193–218, 1985
12. F. Chang, W. Qiu, R. H. Zamar, R. Lazarus, and X. Wang, Journal of Statistical Software, **33**(4):1 – 16, 2010